# DW_fp_square

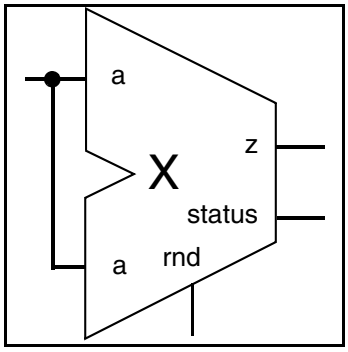
## Floating-Point Square

Version, STAR, and myDesignWare Subscriptions: IP Directory

## Features and Benefits

Revision History

- The precision format is parameterizable for either IEEE single, double precision, or a user-defined custom format
- Hardware for denormal numbers of IEEE 754 standard is selectively provided.
- Fully compatible with the IEEE 754 Floating-point standard with proper set of parameters
- DesignWare datapath generator is employed for better timing and area




## Description

DW_fp_square is a floating-point square logic that squares one floating-point operand: a by a to produce a floating-point product, z.

Component pins are described in Table 1-1 and configuration parameters are described in Table 1-2.

**Table 1-1 Pin Description**

| Pin Name | Width | Direction | Function |
|---|---|---|---|
| a | *exp_width* + *sig_width* + 1 bits | Input | Input data |
| rnd | 3 bits | Input | Rounding mode; supports all rounding modes described in the *Datapath Floating-Point Overview* |
| z | *exp_width* + *sig_width* + 1 bits | Output | Square of a ($a^2$) |
| status | 8 bits | Output | Status flags for the result z<br>For details, see STATUS Flags in the *Datapath Floating-Point Overview.* |

**Table 1-2 Parameter Description**

| Parameter | Values | Description |
|---|---|---|
| sig_width | 2 to 253 bits<br>Default: 23 | Word length of fraction field of floating-point numbers a and z |
| exp_width | 3 to 31 bits<br>Default: 8 | Word length of biased exponent of floating-point numbers a and z |

**Table 1-2 Parameter Description (Continued)**

| Parameter | Values | Description |
|---|---|---|
| ieee_compliance | 0 or 1<br>Default: 0 | Level of support for IEEE 754:<br>■ 0: No support for IEEE 754 NaNs and denormals; NaNs are considered infinities and denormals are considered zeros<br>■ 1: Fully compliant with the IEEE 754 standard, including support for NaNs and denormals<br>For more, see IEEE 754 Compatibility in the *Datapath Floating-Point Overview*. |

**Table 1-3 Synthesis Implementations**

| Implementation Name | Function | License Feature Required |
|---|---|---|
| rtl | Synthesis model | DesignWare |

**Table 1-4 Simulation Models**

| Model | Function |
|---|---|
| DW02.DW_FP_SQUARE_CFG_SIM | Design unit name for VHDL simulation |
| dw/dw02/src/DW_fp_square_sim.vhd | VHDL simulation model source code |
| dw/sim_ver/DW_fp_square.v | Verilog simulation model source code |

DW_fp_square provides the hardware for denormal numbers and NaNs of IEEE 754 standard. If the parameter `ieee_compliance` is turned off, denormal numbers are considered as zeros, and NaNs are considered as Infinity. Otherwise, denormal numbers and NaNs become effective and additional hardware for the square of denormal numbers is integrated.

## Suppressing Warning Messages During Verilog Simulation

The Verilog simulation model includes macros that allow you to suppress warning messages during simulation.

To suppress all warning messages for all DWBB components, define the DW_SUPPRESS_WARN macro in either of the following ways:

- Specify the Verilog preprocessing macro in Verilog code:

  ```
  `define DW_SUPPRESS_WARN
  ```

- Or, include a command line option to the simulator, such as:

  `+define+DW_SUPPRESS_WARN` (which is used for the Synopsys VCS simulator)

The warning messages for this model include the following:

- If an invalid rounding mode has been detected on `rnd`, the following message is displayed:

  ```
  WARNING: <instance_path>:
      at time = <timestamp>: Illegal rounding mode.
  ```

  To suppress this message, use the DW_SUPPRESS_WARN macro explained earlier.

## Related Topics

- Datapath Floating Point Overview
- DesignWare Building Block IP User Guide

## HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DW_Foundation_comp.all;
-- If using numeric types from std_logic_arith package,
-- comment the preceding line and uncomment the following line:
-- use DWARE.DW_Foundation_comp_arith.all;

entity DW_fp_square_inst is
  generic (
    inst_sig_width       : POSITIVE := 23;
    inst_exp_width       : POSITIVE := 8;
    inst_ieee_compliance : INTEGER := 0
  );
  port (
    inst_a      : in std_logic_vector(inst_sig_width+inst_exp_width downto 0);
    inst_rnd    : in std_logic_vector(2 downto 0);
    z_inst      : out std_logic_vector(inst_sig_width+inst_exp_width downto 0);
    status_inst : out std_logic_vector(7 downto 0)
  );
end DW_fp_square_inst;


architecture inst of DW_fp_square_inst is

begin

  -- Instance of DW_fp_square
  U1 : DW_fp_square
  generic map (
    sig_width => inst_sig_width,
    exp_width => inst_exp_width,
    ieee_compliance => inst_ieee_compliance
  )
  port map (
    a => inst_a,
    rnd => inst_rnd,
    z => z_inst,
    status => status_inst
  );

end inst;

-- pragma translate_off
configuration DW_fp_square_inst_cfg_inst of DW_fp_square_inst is
  for inst
  end for;
end DW_fp_square_inst_cfg_inst;
```

```
-- pragma translate_on
```

## HDL Usage Through Component Instantiation - Verilog

```
module DW_fp_square_inst( inst_a, inst_rnd, z_inst, status_inst );

parameter sig_width = 23;
parameter exp_width = 8;
parameter ieee_compliance = 0;

input [sig_width+exp_width : 0] inst_a;
input [2 : 0] inst_rnd;
output [sig_width+exp_width : 0] z_inst;
output [7 : 0] status_inst;

    // Instance of DW_fp_square
    DW_fp_square #(sig_width, exp_width, ieee_compliance)
          U1 ( .a(inst_a), .rnd(inst_rnd), .z(z_inst), .status(status_inst) );

endmodule
```

## Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

| Date | Release | Updates |
|---|---|---|
| July 2020 | DWBB_201912.5 | ■ Adjusted the description of the *ieee_compliance* parameter in Table 1-2 on page 1<br>■ Added "Suppressing Warning Messages During Verilog Simulation" on page 3<br>■ Added this Revision History table and the document links on this page |

# Copyright Notice and Proprietary Information



### Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

### Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

### Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

### Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

### Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com